

Implementation of an FPGA-based actual observer for active suspension control

Laszlo Schaffer

Department of Technical Informatics
University of Szeged
Szeged, Hungary
schaffer.laszlo@inf.u-szeged.hu

Szilveszter Pletl

College of Applied Sciences
Subotica Tech
Subotica, Serbia
pszilvi@vts.su.ac.rs

Zoltan Kincses

College of Applied Sciences
Subotica Tech
Subotica, Serbia
zoltan.kincses@vts.su.ac.rs

Abstract—In the case of vehicle automation the central role is played by the security, the reliability and the convenience. The elements used in a suspension system with fixed parameter values do not provide the desired safety and comfort conditions in all circumstances. In this paper an active suspension system is presented, which is capable of minimal sensor information based efficient intervention. The proposed architecture enables a cost-effective and energy efficient implementation of an actual observer. In this work the mathematical model, the analysis of the model, the measured variables, the design of the actual observer and the Field Programmable Gate Array (FPGA) implementation of the actual observer is presented. The accuracy and the correct operation of the proposed FPGA-based actual observer controlled active suspension system are validated by simulations.

Keywords— *suspension control; actual observer; FPGA; control theory*

I. INTRODUCTION

The most significant part, which affects the handling performance and comfortability of a vehicle is the suspension system. A traditional suspension system consists of springs and dampers to decrease road noise, bump effects and vibrations. There are three types of suspension systems, which are the passive, the semi-active and the active ones.

Nowadays vehicles use passive suspension system, which consists of springs, that can store energy and dampers for energy dissipation. In a passive suspension system the spring and damper coefficients are constants, and can not adapt to the road surface in all circumstances.

In addition to the passive type a semi-active suspension system is capable of controlling the damping coefficient. It can offer an improvement in the dynamic performance of the vehicle, and more effective than the passive suspension, but unfortunately can not handle the high frequency road noise. The Electro Rheological (ER) and Magneto Rheological (MR) fluid dampers are two types of semi-active suspension systems [1].

As the sensor, signal processor and electro-hydraulic component technology upgraded the active suspension system was born. An active suspension system uses actuators to add force, rather than dissipating energy. It can create a restoring force to maintain the steadiness of the vehicle, dampen the vibration and avoid the lifting of the wheels. Based on the

sensor values the control algorithm controls the additional force to pull down or to push up the body masses, thus it can compensate the disturbances of the road. The drawback is the actuation energy needed for the compensation, because an external energy source is required, which increases the system cost. It is proved, that an active suspension system can maintain good comfort and handling performance in various conditions [2].

Reviewing the literature of active suspension system control various approaches can be found, such as sliding [3], Linear-Quadratic Regulator (LQR) [4], and PID control [5]. Although all of them is capable of an effective active suspension control, in this work the LQR based controller is used, because it has easier implementation and has similar efficiency.

In an active suspension system, some state variables can not be measured directly and the usage of sensors is not necessary, due to the cost, accuracy and reliability. Therefore a full state feedback control can not be used, unless a state observer is applied. In [3] the road profile is estimated via a disturbance observer, and the active suspension system is controlled by a sliding mode control algorithm. A sliding mode controller and a sliding mode observer is used together in [6]. The observer can detect and isolate the failure of sensors, because every fault has its own pattern, and can be recognized.

Due to the great flexibility, high computational capacity and boundless reconfiguration possibilities of Field Programmable Gate Arrays (FPGA), FPGAs can be used to easily implement various types of controlling algorithms. In the literature of FPGA-based active suspension control some approaches can be found. In [7] a Piece-Wise Multilinear Fuzzy Inference System (PWM-FIS) controlled semi-active suspension system is presented. It is proven that the FPGA architecture is capable of real-time advanced control of any suspension system. An Adaptive Neuro Fuzzy Inference System (ANFIS) is used for controlling a nonlinear active suspension system in [8]. Very High speed integrated circuit Hardware Description Language (VHDL) is used for the implementation, and offline training is needed for the controller. It provides an efficient and robust hardware implementation of the control algorithm.

Based on the literature review of Field Programmable Gate Array (FPGA) based observers some approaches can be found.

TABLE I. ZYNQ-Z7020 GENERAL AND DEDICATED RESOURCES

Resource type	Name	Quantity
General resources	Slices	13,300
	6-input LUT	53,200
	Flip-Flops	106,400
Dedicated resources	36Kbit RAM	140
	DSP48E1 Slice	220
Microprocessor	ARM-Cortex A9	2 Cores

In [9] a linear observer is designed and implemented on a Virtex-5 FPGA, while in [10] the design and implementation of a sliding-mode observer is presented.

Currently in the literature active suspension system control through FPGA-based observer can not be found, so it is reasonable to implement an actual observer on an FPGA for real-time observer based control. The easy reconfiguration of FPGAs gives opportunity for education purposes, because students can analyze the operation of observer based control, and study the effect of parameter changes, without the risk of damaging or endangering the vehicle or the suspension system.

II. FIELD PROGRAMMABLE GATE ARRAYS

The FPGAs are programmable semiconductor devices, which consists of a matrix of programmable logic elements connected through programmable interconnections. The basic structure of a Xilinx FPGA is built up from Configurable Logic Blocks (CLBs), a Programmable Interconnection (PI) network, Block RAM memories (BRAMs), configurable I/O Blocks (IOBs), Digital Signal Processors (DSP) and Digital Clock Management (DCM) blocks. The CLBs are the main digital processing blocks, which are capable of performing combinational as well as sequential operations. For complex arithmetic operations DSP blocks can be used, while the BRAM resources can be used for internal data storage purposes. FPGA architectures can contain embedded soft or hard core microprocessors (MicroBlaze, ARM processor cores), support numerous data communication protocols (USB, Ethernet, SPI, I2C, UART, etc.) and additional peripherals. [11]

III. DIGILENT ZEDBOARD

The actual observer was implemented on a ZedBoard which has a Zynq-z7020 (XC7Z020) FPGA (Programmable Logic – PL), and an embedded ARM Cortex A9 hard core microprocessor (Processing System – PS). The XC7Z020 FPGA is based on the Artix-7 FPGA architecture. The general and dedicated resources of this device are shown in Table 1. One CLB of the Artix-7 FPGA contains two Slices. A Slice contains eight registers, four 6-input LUTs, multiplexers and carry logics as general logic resources. A single LUT can be configured as a 32-bit shift register, a 6-input LUT or a 64-bit distributed RAM. Dedicated resources can be found on the Artix-7 architecture like 36kbit dual port BRAM memories, and 25x18bit DSP Slices. The ARM Cortex A9 microprocessor can work with a maximum of 866 MHz clock frequency, has 32 KB instruction and 32 KB data L1 cache in one core, 512

TABLE II. PARAMETERS OF THE ACTIVE SUSPENSION SYSTEM MODEL

Parameter	Description	Value	Unit
M_1	Quarter mass of the vehicle	250	kg
M_2	Suspension mass	320	kg
K_1	Spring coefficient of the Suspension	80000	N/m
K_2	Spring coefficient of the Wheel	500000	N/m
b_1	Damping coefficient of the Suspension	350	Ns/m
b_2	Damping coefficient of the Wheel	15020	Ns/m
U	Control force	-	Nm
X_1	Position of the vehicle body	-	cm
X_2	Position of the suspension	-	cm
r	Disturbance from the road	-	cm

KB L2 cache memory, and 256 KB on chip memory. It supports DDR3 external memory, UART, CAN, I2C, SPI communication and maximum of four GPIO peripherals. The microprocessor communicates with the PL through AXI buses, and 16 interrupt ports.

IV. ACTIVE SUSPENSION SYSTEM MODEL

Only the quarter of the active suspension system has been modelled and implemented in MATLAB/Simulink, because of an easier implementation. The model consists of two masses, the body and the suspension mass, which are connected to each other through a spring, a damper and a controllable actuator.

The suspension mass is connected to the road through a spring and a damper. A quarter of the active suspension system model can be seen on Fig. 1, and the parameters of the system are shown in Table 2.

The mathematical model of the system can be obtained using Newton's law:

$$\ddot{X}_1 = \frac{1}{M_1} \cdot (-b_1 \cdot \dot{X}_1 + b_1 \cdot \dot{X}_2 - K_1 \cdot X_1 + K_1 \cdot X_2 + U) \quad (1)$$

$$\ddot{X}_2 = \frac{1}{M_2} (b_1 \cdot \dot{X}_1 - b_1 \cdot \dot{X}_2 + K_1 \cdot X_1 - K_1 \cdot X_2 + \quad (2)$$

$$b_2 \cdot \dot{r} - b_2 \cdot \dot{X}_2 + K_2 \cdot r - K_2 \cdot X_2 - U)$$

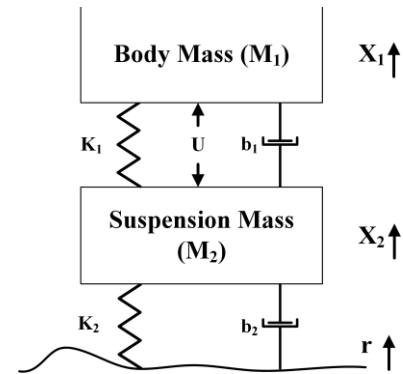


Fig. 1. A quarter of the active suspension system model

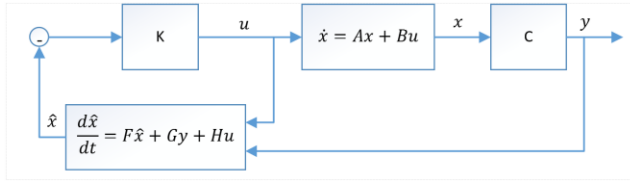


Fig. 2. The flow diagram of an actual observer based system

It can be seen, that the control force (U) and the disturbance from the road (r) are the two inputs of the system, while the output of the active suspension system is the distance between the vehicle body and the suspension ($X_1 - X_2$).

Using (1) and (2) the state-space model can be written in the form of:

$$\dot{x} = A \cdot x + B \cdot u, \quad y = C \cdot x + D \cdot u \quad (3)$$

Where x is the state vector, which contains the position and the velocity of the vehicle body and the suspension. It can be written as follows:

$$x = \begin{bmatrix} X_1 \\ \dot{X}_1 \\ X_2 \\ \dot{X}_2 \end{bmatrix} \quad (4)$$

Using (1) and (2) in the form of (3) the state-space matrices can be expressed:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-k_1}{M_1} & \frac{-b_1}{M_1} & \frac{k_1}{M_1} & \frac{b_1}{M_1} \\ 0 & 0 & 0 & 1 \\ \frac{k_1}{M_2} & \frac{b_1}{M_2} & \left(\frac{-k_1 - k_2}{M_2}\right) & \left(\frac{-b_1 - b_2}{M_2}\right) \end{bmatrix} \quad (5)$$

$$B = \begin{bmatrix} 0 \\ \frac{1}{M_1} \\ 0 \\ \left(-\frac{1}{M_2}\right) \end{bmatrix} \quad (6)$$

$$C = [1 \quad 0 \quad -1 \quad 0] \quad (7)$$

It can be seen that the D matrix is not presented, because it contains zeros. The active suspension system modelled in the state-space form is the base of the actual observer, and the observer is not prepared for the road disturbance, therefore the r input has been removed.

A model based on the equation system of (1) and (2) and a model in the state-space form have been implemented using the MATLAB/Simulink environment. The equation based model can be seen in Fig. 2, which will be the plant, that is controlled by the observer based LQR control.

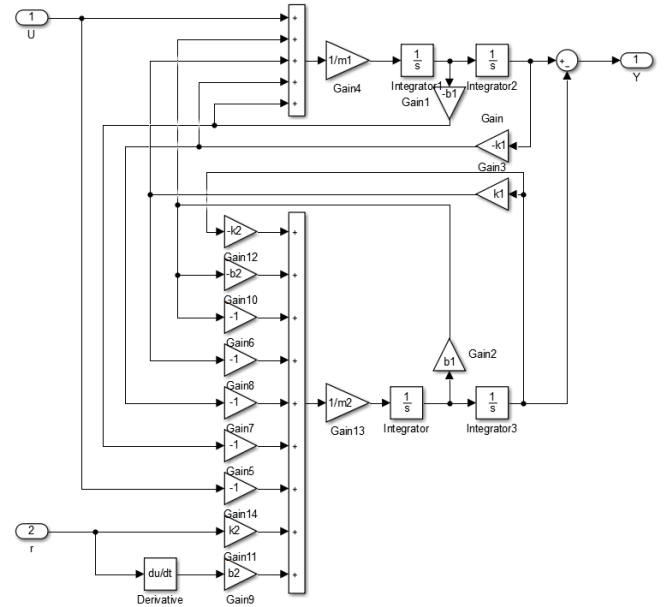


Fig. 3. A Simulink implementation of the active suspension model

V. THE ACTUAL OBSERVER

In a full-state feedback control all state variables should be measurable, if not the usage of an observer is required. Based on the output of the plant the actual observer can estimate the states and use them for a full-state feedback control. The main equation of the actual observer can be written as follows:

$$\frac{d\hat{x}}{dt} = F\hat{x} + Gy + Hu, \quad F = A - GCA, \quad (8)$$

where \hat{x} is the estimated state vector, F , G and H are the observer gains. The dimension of F is 4×4 , while G and H are only 4×1 vectors. The flow diagram of an actual observer based system can be seen in Fig. 3, where a simple full-state feedback control is used, based on the estimated states.

The actual observer works in discrete time, so the eigenvalues of the G matrix should be placed inside the unit circle with pole placement. The poles chosen for the G matrix are:

$$\text{poles}(G) = [0.63 \ 0.62 \ 0.61 \ 0.6]$$

The Simulink model of the active suspension system can be seen on Fig. 4., where the *Real Suspension System* is based on (1) and (2). The *Rate Transition* blocks are converting the continuous time signals to discrete time, while the *LQRK* variable is the gain of the LQR control.

The Linear Quadratic Regulator is based on the following quadratic cost function:

$$J = \int_0^\infty (x^T Q x + u^T R u) dt, \quad (9)$$

where the matrix Q is symmetric positive semi-definite and R is positive symmetric definite. In this case the optimal linear feedback control law can be obtained as:

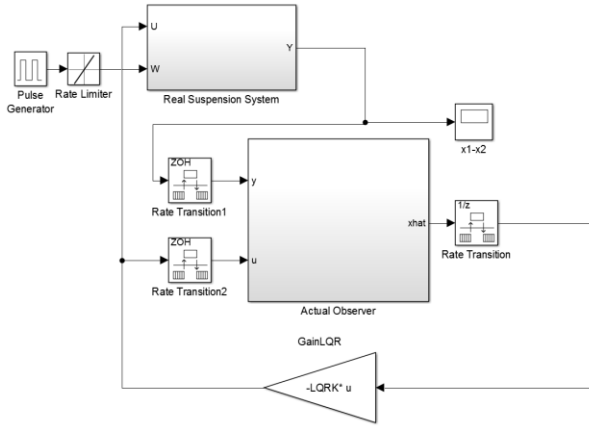


Fig. 4. The active suspension control Simulink system

$$u(t) = -Kx(t), \quad (10)$$

The weighting matrices Q and R are:

$$Q = \begin{bmatrix} 100000 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = 4 \cdot 10^{-6}$$

VI. FPGA-BASED ACTUAL OBSERVER

The implementation of the FPGA-based actual observer was made in the Vivado High Level Synthesis (HLS) programming environment, which enables fast prototyping with the usage of C/C++ code and enables the export of the resulting Intellectual Property (IP) Core into the Xilinx System Generator, so it can be used in the Simulink.

As in (8) can be seen the estimated state of the observer can be calculated with three matrix multiplication, and a summation. The values of the previous state estimation should be stored in registers and used in the multiplication with the F matrix. Using this technique, a delay can be made, so (8) will be satisfied. The F , G , H matrices are pre-calculated, and stored as constants. In the Simulink system, the *Actual Observer* block is replaced with the generated HLS actual observer IP core, and with Co-Simulation through JTAG, it can be evaluated in the FPGA. The block diagram of the FPGA-based actual observer can be seen in Fig.5.

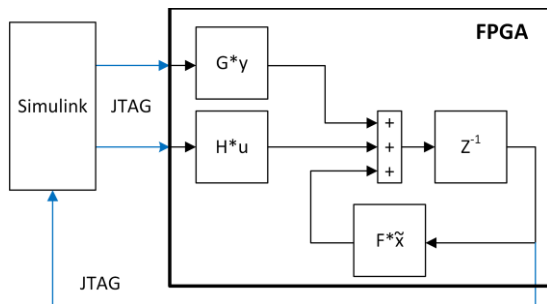


Fig. 5. The active suspension control Simulink system

TABLE III. RESOURCE REQUIREMENTS OF THE ACTUAL OBSERVER

Resource type	Name	Quantity	Utilization (%)
General resources	Slices	218	1.6
	6-input LUT	567	1
	Flip-Flops	610	0.6
Dedicated resources	36Kbit RAM	0	0
	DSP48E1 Slice	32	14

VII. RESULTS

The active suspension system was tested either using Simulink-based simulation and real FPGA hardware. The results show that the actual observer can be implemented on the ZedBoard with an 8.47 ns clock period, maximum of 100 clock cycle latency, which means one calculation takes $0.8 \mu s$. The DSP Slice requirement is the function of the number of add/sub and multiplication blocks. Because the calculations of (8) includes only three matrix multiplications, and two of them can be done at the same time, the architecture uses 32 DSP slices. The constant matrices are stored in LUTs, so no BRAM is required. The resource requirements can be seen in Table 3.

VIII. CONCLUSION AND FUTURE WORK

It can be concluded, that the actual observer can be implemented on an FPGA, and it can calculate the results in real-time, with 8.47 ns clock period, $0.8 \mu s$ per calculation. The Simulink has the simulation period set to 0.001, therefore the observer calculation can run 125 times faster, than the Simulink.

In the future other observer and control algorithms can be implemented in this framework, so it can be used not only for experimental, but for education purposes as well. Students can analyze the operation of active suspension system, and study the effect of different parameter changes, without the risk of damaging or endangering the real physical system. Furthermore, the plant and the corresponding model can be changed to other interesting models.

REFERENCES

- [1] B. L. J. Gysen, J. L. G. Janssen, J. J. H. Paulides, E. A. Lomonova, "Design aspects of an active electromagnetic suspension system for automotive applications", Industry Applications Society Annual Meeting, IEE, 2008
- [2] M. A. Nekoui, P. Hadavi, "Optimal control of an active suspension system", Power Electronics and Motion Control Conference, 2010
- [3] E. Akbari, M. Farsadi, I. Z. M. Darus, R. Ghelichi, "Observer Design for Active Suspension System Using Sliding Mode Control", IEEE Student Conference on Research and Development, 2010, pp. 207-212
- [4] A. Agharkakli, G. S. Sabet, A. Barouz, "Simulation and Analysis of Passive and Active Suspension System Using Quarter Car Model for Different Road Profile", International Journal of Engineering Trends and Technology, Volume 3, Issue 5, 2012, pp. 636-644
- [5] M. Avesh, R. Srivastava, "Modeling Simulation and Control of active suspension system in Matlab Simulink environment", Students Conference On Engineering and Systems, 2012
- [6] A. Chamseddine, H. Noura, "Control and Sensor Fault Tolerance of Vehicle Active Suspension", IEEE Transactions on Control Systems Technology, Volume 16, Issue 3, 2008, pp. 416-433

- [7] K. Basterretxea, I. del Campo, J. Echanobe, “A semi-active suspension embedded controller in a FPGA”, Symposium on Industrial Embedded Systems, 2010
- [8] A. A. Aldair, W. Wang, “FPGA based adaptive neuro fuzzy inference controller for full vehicle nonlinear active suspension systems”, International Journal of Artificial Intelligence & Applications, Volume 1, Number 4, 2010, pp. 1-15
- [9] R. S. Rana, B. Kumari, “FPGA Implementation of Linear Observer”, International Journal of Future Computer and Communication, Volume 3, Number 1, 2014, pp. 14-17
- [10] A. Lienhardt, G. Gateau, T. A. Meynard, “Digital Sliding-Mode Observer Implementation Using FPGA”, IEEE Transactions on Industrial Electronics, Volume. 54, Number 4, 2007, pp. 1865-1877
- [11] H. Yang, J. Zhang, J. Sun, L. Yu, „Review of Advanced FPGA Architectures and Technologies”, Journal of Electronics, Volume 31, Issue 5, 2014, pp. 371-393